Tree-based prediction on incomplete data using imputation or surrogate decisions

Holger Cevallos Valdiviezo[1] and Stefan Van Aelst[2,1]

[1] Ghent University, Department of Applied Mathematics, Computer Science and Statistics, Krijgslaan 281 S9, Gent, Belgium

[2] KU Leuven, Department of Mathematics, Section of Statistics, Celestijnenlaan 200B B-3001, Leuven, Belgium

Abstract

The goal is to investigate the prediction performance of tree-based techniques when the available training data contains features with missing values. Also the future test cases may contain missing values and thus the methods should be able to generate predictions for such test cases. The missing values are handled either by using surrogate decisions within the trees or by the combination of an imputation method with a tree-based method. Missing values generated according to missing completely at random (MCAR), missing at random (MAR) and missing not at random (MNAR) mechanisms are considered with various fractions of missing data. Imputation models are built in the learning phase and do not make use of the response variable, so that the resulting procedures allow to predict individual incomplete test cases. In the empirical comparison, both classification and regression problems are considered using a simulated and real-life datasets. The performance is evaluated by misclassification rate of predictions and mean squared prediction error, respectively. Overall, our results show that for smaller fractions of missing data an ensemble method combined with surrogates or single imputation suffices. For moderate to large fractions of missing values ensemble methods based on conditional inference trees combined with multiple imputation show the best performance, while conditional bagging using surrogates is a good alternative for high-dimensional prediction problems. Theoretical results confirm the potential better prediction performance of multiple imputation ensembles.

Keywords: prediction, missing data, surrogates, multiple imputation, conditional inference tree

1 Introduction

Many real datasets with predictive applications face the problem of missing values on useful features. Evidently, this complicates the predictive modeling process since predictive power may depend heavily on the way missing values are treated. In principle, missing data can occur in the training data only, in the individual test cases only, or in both the training data and test cases. In practice, however, missing data appear most often in both training and test set. Consider for instance

customer data that is used to predict important outcomes such as buying preferences for individual costumers (based on their past actions). This type of data frequently contains missing values in both the training data and test cases, because the same amount of information is not available for all customers.

Most of the research work so far has addressed the problem of missing values in the training data (see e.g. [2, 9, 13, 16, 34, 37]). On the other hand, [36] is one of the only contributions in which the prediction accuracy of classification techniques is compared when only test cases contain missing values. Tree-based classifiers have been investigated for test cases with data missing completely at random (MCAR), i.e. test cases with missingness which does not depend on any value of the data. The performance of prediction methods for different missing data strategies when missing data occur in both the training and test set has been assessed in [15, 22, 32]. However, in [32] k-nearest neighbors (kNN) imputation was applied separately on the training and test samples. This is a potential weakness for practical purposes because the kNN imputation is impossible for test cases that appear on a case-by-case basis. Similarly, in [15] and [22] imputation models were applied separately to the training and test cases. Moreover, the response variable was used in the imputation model for the training data so that the same imputation scheme cannot be applied to test cases arriving one-by-one. In this study, we are interested in methods that can deal with missingness in both training and test cases. Moreover, the methods should be able to handle test cases that appear one-by-one, because this case is often encountered in practical applications. Think for example of new potential patients for which a prediction needs to be made as soon as possible on a case-by-case basis, using the available information of the patient (such as clinical test results).

In this work we compare several strategies to handle missing data when using tree-based prediction methods. We focus on trees because they have several advantages and few limitations compared to other prediction techniques. Firstly, trees allow to handle data of different type (categorical, discrete, continuous). Other features that make trees highly popular among practitioners are their ability to capture important dependencies and interactions. Moreover, tree-based ensembles such as random forests can easily handle high dimensional problems and often show good performance without the need to fine-tune parameters. Trees also include a built-in methodology to process observations with missing data, called surrogate splits [6].

Evidently, if the missing data issue is not addressed correctly, misleading predictions may be obtained. Thus, one aims for prediction rules that have low bias (accurate enough) and low variability (stable enough) and at the same time take into account the additional uncertainty caused by missing values. Among the strategies to handle the missing values are:

1. Discard observations with any missing values in the training data
2. Rely on the learning algorithm to deal with missing values in the training phase
3. Impute all missing values before training the prediction method

Approach 1 encompasses ad-hoc procedures like complete case and available case analysis. They have been shown to work for relatively small amounts of missing data and under certain restrictive conditions [44, 48]. However, this approach is not applicable when missing values are present in test cases. Tree methods with surrogate splits are an example of the second approach. An advantage of strategy 2 is that incomplete data need not be treated prior to model fitting. For most learning techniques, the third approach is necessary to handle incomplete values or it simply helps to improve predictive capability. Many imputation methods have been developed to address the missing data issue in general. Imputation methods have been studied extensively with regard

to inference: unbiasedness of estimates, efficiency, coverage and length of confidence intervals or power of tests (see e.g. [8, 11, 26, 38]). Other works study the performance of imputation methods when estimating the true values of the missing data, without considering the subsequent statistical analysis (see e.g. [24, 39]). However, there is much less known about the properties of imputation methods in the context of prediction. An advantage of Approach 3 is that it completely separates the missing data problem from the prediction problem. This strategy thus gives freedom to (third party) analysts to apply any appropriate data mining method to the imputed data.

A few comparisons of approach 2 and 3 have already been considered in the literature. For instance in [13] CART using surrogates was compared to CART preceded by single or multiple imputation. Two classification problems were considered. Multiple imputation performed clearly better than both single imputation and surrogates. Single imputation outperformed surrogates for a fraction of missingness above 10%. No ensemble methods were considered.

The predictive performance of conditional random forests [20] with missing data was investigated in [32]. Conditional random forests (CondRF) combined with surrogates was compared to CondRF with prior kNN imputation. Both classification and regression problems were considered. No difference in performance was found between handling missing values by surrogates or with prior kNN imputation. Recently, [15] compared the predictive performance of CART, conditional inference tree (CondTree) and CondRF in combination with surrogates or Multiple Imputation by Chained Equations (MICE) to handle the missing data. Real datasets with and without missing cells were used. The complete data were used for a simulation study in which missing values were introduced completely at random. For the real data with missing values MICE did not show a convincing improvement compared to surrogates, while in their simulation study MICE was beneficial for large amounts of missing data introduced in many variables. However, the authors argue that their simulation results may lack generalizability due to restrictive and artificial simulation patterns. Therefore, it is suggested to extend their simulations to a wider range of patterns.

So far, there is no clear conclusion in the literature about which combinations of tree-based prediction method and missing data strategy yield the most satisfactory predictions. It seems that an answer to this question may depend on the structure of the predictors, the type of relationship between predictors and response variable, and the pattern and fraction of missing data.

The contribution of this paper is threefold. First, we provide a theoretical comparison of prediction techniques that can be constructed from incomplete training data and can be applied directly on individual test cases with missing values, as this corresponds to most of the practical applications. Secondly, we set up a framework for the empirical comparison of these prediction techniques. Thirdly, using this framework, we provide some insight into the effect of different missing data patterns on the performance of 26 of these techniques based on trees.

In our comparison we consider as learning methods CART, CondTree, Random Forest (RF), CondRF, Bagging and Conditional Bagging (CondBagging). The procedures to handle missing data are surrogates, single imputation by median/mode, proximity matrix or kNN, and multiple imputation by MICE or Multiple Imputation by Sequential Regression Trees (MIST). Not all combinations have been implemented in R [31] which we use for our investigation. The 26 techniques in our comparison are summarized in Table 1.

Our comparison incorporates recent tree-based methods and imputation procedures for which there are almost no research results available about their predictive performance in presence of missing values. Any analysis or discussion of the situations under which the different techniques predict well or poorly is still lacking. Our empirical comparison shows that for moderate to large amounts of missing data, multiple imputation by MICE or MIST followed by CondRF is advisable,

although these techniques are expensive in terms of computation time. Their better performance is due to the mutual effort of the imputation strategy and prediction method to average out sampling variability and variability due to missing data. This result of our empirical comparison is confirmed by the theoretical derivations. CondBagging using surrogate decisions emerges as an alternative with good performance and much lower computation time. For small amounts of missing data, any ensemble method with surrogate decisions or preceded by single imputation suffices to get a good prediction performance at a cheaper computational cost.

Table 1: Overview of the 26 techniques investigated in this study. Each mark '×' corresponds to a technique. The second mark in the MIST + RF box corresponds to a special case of this technique that consists of imputing bootstrap samples by MIST + RF. N/I stands for "not implemented".

Strategy for miss. data	Imputation method	CART	CondTree	RF	CondRF	Bagg.	CondBagg.
Surrogates	None	×	×	N/I	×	×	×
Single Imp.	Median/mode	×	×	×	×	N/I	N/I
	Prox.matrix	×	×	×	×	N/I	N/I
	kNN	×	×	×	×	N/I	N/I
Multiple Imp.	MICE	×	×	×	×	N/I	N/I
	MIST	×	×	××	×	N/I	N/I

2 Methodology

2.1 Tree-based methods

The Classification and Regression Tree (CART) algorithm proposed by [6] is a popular technique to fit trees. While it is an intuitively appealing procedure, it also has some drawbacks: it is known to be highly unstable due to its hierarchical nature [17, 28] and it tends to produce selection bias towards continuous and categorical features with many possible splits and missing values. Aiming to solve the latter problem, [21] proposed the conditional inference tree (CondTree) algorithm which utilizes a unified framework for conditional inference. More specifically, CondTree allows for unbiased selection of the splitting variable by using univariate P-values which can be directly compared among covariates measured at different scales. However, CondTree might still be an unstable procedure due to its hierarchical nature.

With the aim of reducing the prediction variance of single trees, Bagging was proposed [3]. It fits the noisy CART algorithm many times to bootstrap-sampled versions of the data [12] and averages for each observation the outcomes of individual trees to obtain a final prediction. However, overfitting may arise because trees are fitted on modified versions of the same original sample. This limits the benefits of Bagging. Hence, Random Forest [5] was developed to further improve the prediction variance reduction of Bagging by decreasing the correlation among trees. This is established by adjusting the splitting process during the growing of the tree. Instead of considering all features for each split, only a number $g \leq p$ of predictors selected at random are considered as candidates for a split.

In the same spirit, Conditional Bagging and Conditional inference Forests were developed to combine the benefit of unbiased variable selection with reduction of the prediction variance [20].

Surrogate splits, as introduced in [6], are an attempt to mimic the primary split of a region in terms of the number of cases sent down the same way. For any observation with a missing

value for the primary split variable, we can find among all variables with nonmissing value for that case the predictor and corresponding split point producing the best surrogate split (i.e. the split yielding the most similar results as the best split). [30] considers surrogate splits as a special case of predictive value imputation. All tree-based methods can in theory handle missing predictor values by using the principle of surrogate splits. However, the implementation of RF in the R package randomForest [25] cannot be used on incomplete data. More information about tree-based methods is given in the supplementary material.

2.2 Imputation methods

An imputation can be the mean or a random draw from a predictive distribution that is specifically modeled for each missing entry [26]. Thus, an imputation method is required to estimate these predictive distributions based on the observed data. In general, an advantage of using an imputation strategy is that it separates the missing data problem from the prediction problem. Hence, a completed dataset(s) can be used for the prediction problem. This allows to apply the most appropriate prediction method on the imputed dataset(s). We now give a short description of the imputation methods used in this paper.

Single imputation (SI) methods

A rapid and simple fix to the problem of missing predictor values consists of just replacing them with the column median or mode, depending on the type of predictor variable. However, this method might distort the covariate distribution by underestimating its variance and also the relations between the covariates may be disturbed.

A more elaborate method consists of imputing based on the *proximity matrix* [25], which is a $N \times N$ matrix (N being the size of the training sample) that comes "for free" in the output of the Random Forest implementation in R. Each cell of this matrix contains the proportion of the total number of trees in the forest in which the respective pair of training observations share a terminal region. The proximity matrix algorithm starts with a median/mode imputation. Then, Random Forest is called with the completed data. The imputed values are updated according to the current proximity matrix. For continuous predictors the imputation update is the weighted average of the initially non-missing observations, where the weights are the proximities. For categorical predictors the imputation update is the category with the largest average proximity. This process is repeated iteratively, usually five times. Thus, the intuitive idea is to give a larger weight to cases that are more like the case with missing data.

Another single imputation method is kNN imputation [43]. This procedure looks for the k nearest neighbors of the missing observation with respect to their Euclidean distance computed from the remaining observed variables. Eventually, the missing value is replaced by a weighted mean of the k nearest neighbors, where the weights are based on the kNN euclidean distances.

After imputation by a SI method, the filled-in data are treated as if they were actually observed. The additional uncertainty caused by missing data on top of the already "available" sampling variance is thus ignored. As a consequence, the whole prediction rule may lose stability and hence prediction performance.

Multiple imputation (MI) methods

One way to take into account the variability caused by missing data is through multiple imputations [34, 35]. This creates several training datasets differing only in the imputed fields. The variability across these completed versions of the data reflects the uncertainty underlying the imputed values.

Let M denote the data matrix and D the total number of imputed datasets by MI. As described in [26], multiple imputation draws the missing values for the oth imputed dataset ($o = 1, \ldots, D$) as:

$$M_{\text{mis}}^{(o)} \sim Pr(M_{\text{mis}}|M_{\text{obs}}), \tag{1}$$

with

$$Pr(M_{\text{mis}}|M_{\text{obs}}) = \int Pr(M_{\text{mis}}|M_{\text{obs}}, \boldsymbol{\theta}) Pr(\boldsymbol{\theta}|M_{\text{obs}})\, \mathrm{d}\boldsymbol{\theta}. \tag{2}$$

That is, the imputed values are random draws from the joint posterior distribution of the missing data given the observed data. However, it is often difficult to draw from this predictive distribution due to the requirement of integrating over the model parameters $\boldsymbol{\theta}$ in (2). In the univariate case, Data Augmentation [40] accomplishes this by iteratively drawing a sequence of values of the parameters and missing data until convergence. More specifically, data augmentation can be run independently D times to generate D iid draws from the approximate posterior distribution involving D estimates $\boldsymbol{\theta}^{*(1)}, \boldsymbol{\theta}^{*(2)}, \ldots, \boldsymbol{\theta}^{*(D)}$ from $Pr(\boldsymbol{\theta}|M_{\text{obs}})$ which are subsequently used in the conditional distributions $Pr(M_{\text{mis}}|M_{\text{obs}}; \boldsymbol{\theta}^{*(o)})$ to draw D imputations. However, in situations with multivariate data involving nonlinear relationships, building one coherent model for the joint distribution of the variables may be difficult. In those situations, simpler methods that approximate draws from (1) should be considered. We now discuss two such methods which are used in our comparison.

Multivariate Imputation by chained equations (MICE)

In real multivariate settings with more than one variable containing missing values, we might be able to approximate draws from (1) by specifying for each incomplete variable a conditional model for the missing data given a set of other variables. Essentially, for each variable containing missing values MICE draws values for the parameters and imputations from the corresponding conditional model and iterates this procedure through the other incomplete variables. Hence, the procedure splits the p-dimensional problem into p one-dimensional problems. By modeling only conditional distributions many complexities of real-life multivariate data such as predictors of different type, existence of nonlinear relations or interactions between variables and circular dependence can be addressed [8, 11, 38, 45]. These complexities are difficult to handle if a joint modeling approach [37] is adopted. The reason is that in joint modeling an explicit multivariate distribution for the missing data needs to be specified to derive conditional models for imputations. Thus, distributional assumptions are imposed which may lack flexibility to address the above mentioned complexities. On the other hand, MICE (also called fully conditional specification [FCS] by [46]) directly specifies conditional models without the need of an explicit multivariate model for the entire dataset. Instead, the algorithm assumes that an underlying multivariate model exists and that draws from it can be generated by iteratively sampling from the conditionally specified imputation models.

Let $\mathbf{X}$ be the $N \times p$ matrix that contains the partially observed values for the p predictor variables. Then, $Pr(\mathbf{X}_{\text{mis}}|\mathbf{X}_{\text{obs}})$ denotes the joint *multivariate* posterior where $\mathbf{X}_{\text{mis}}$ and $\mathbf{X}_{\text{obs}}$ are

the missing and observed parts of $\mathbf{X}$, respectively. Assume that the multivariate distribution of $\mathbf{X}$ is completely specified by $\boldsymbol{\theta}$, a p-dimensional vector of unknown parameters. MICE aims to obtain the posterior distribution of $\boldsymbol{\theta}$ through chained equations which form parametric models for the conditional distributions. More precisely, if all p predictors contain missing data, then starting from a simple draw from the observed marginal distributions the tth iteration of chained equations is a Gibbs sampler that successively draws:

$$\begin{aligned}
\theta_1^{*(t)} &\sim Pr(\theta_1 | x_1^{\text{obs}}, x_2^{t-1}, \ldots, x_p^{t-1}) \\
x_1^{*(t)} &\sim Pr(x_1^{\text{mis}} | x_1^{\text{obs}}, x_2^{t-1}, \ldots, x_p^{t-1}, \theta_1^{*(t)}) \\
&\vdots \\
\theta_p^{*(t)} &\sim Pr(\theta_p | x_p^{\text{obs}}, x_1^{t}, x_2^{t}, \ldots, x_{p-1}^{t}) \\
x_p^{*(t)} &\sim Pr(x_p^{\text{mis}} | x_p^{\text{obs}}, x_1^{t}, x_2^{t}, \ldots, x_{p-1}^{t}, \theta_p^{*(t)}),
\end{aligned} \tag{3}$$

where $x_j^{(t)} = (x_j^{\text{obs}}, x_j^{*(t)})$ is the jth imputed feature at iteration t and $\theta_1, \ldots, \theta_p$ are the components of $\boldsymbol{\theta}$ (see [46, 47]).

MICE deviates from Markov Chain Monte Carlo (MCMC) approaches in that the sequences of univariate regressions are applied to cases with observed x_j. After convergence, it is implicitly assumed that the Gibbs sampler in (3) provides a draw $\boldsymbol{\theta}^*$ from its posterior which can be used to draw values $\mathbf{X}^*$ to impute $\mathbf{X}_{\text{mis}}$. [47] states that convergence of the algorithm can be quite fast (10 iterations might be enough) since previous imputations $x_j^{*(t-1)}$ only enter $x_j^{*(t)}$ through their relation with other variables. This procedure can be run in parallel D times to generate D imputations. Various authors have shown the satisfactory performance of this method in a variety of simulation studies (e.g. [16, 19, 46]). As mentioned earlier, MICE also gives the user flexibility to specify a convenient imputation model for each variable in order to help preserving important characteristics of the data. Due to its construction, this approach is suitable for data missing at random (MAR), i.e. data whose missingness depends only on the observed data, although [47] argues that MICE can also handle data missing not at random (MNAR) under additional modeling assumptions. Data MNAR occur when the missingness depends on unobserved data.

Despite the mentioned benefits, the MICE algorithm also has some shortcomings. For instance, it is not guaranteed that the specified conditional models in the Gibbs sampler will eventually converge to an existing stationary distribution. This problem is known as incompatibility of the conditionals which however is not considered a serious problem in practice [46]. Another issue is that the standard MICE implementation uses parametric (generalized) linear models to estimate the conditional distributions in (3). Therefore, it might not be able to capture complex relations among variables, especially when having a large number of predictors.

Multivariate Imputation by Sequential Regression Trees (MIST)

MIST has been proposed in [8] with the goal of better capturing interactions and nonlinear relations among predictors when imputing missing values. MIST uses CART to model the conditional distribution of each missing predictor in (3). The authors justify their choice for CART by stressing that it is sufficiently flexible to capture complex structures without parametric assumptions or data transformations. After convergence, approximate draws from the predictive distribution of

the incomplete targeted predictor can be taken by sampling elements from the final region that corresponds to the covariate values of the case of interest. A Bayesian bootstrap [33] is performed within each final region before sampling in order to reflect the uncertainty about the population conditional distributions [8]. Another benefit of this strategy is that potential problems that may arrive when imputing, such as nonsensical or impossible imputations, are avoided because MIST imputations come from the observed values.

There are two sources of uncertainty that might prevent us to produce good prediction results when using data with incomplete features: one is the inherent sampling variability and the other is the additional uncertainty caused by missing data. The former is well-known and can affect the performance of highly data-driven prediction methods such as single tree methods. The latter can make the prediction rule unreliable if not treated adequately, even if the prediction method itself is very stable. For instance, if the imputation is poor then the predictions can become unreliable no matter how well the learning method performs. This can happen when applying a single imputation prior to the learning method.

Procedures that combine MI with an ensemble of trees might potentially yield superior results, thanks to the mutual effort of the imputation strategy and prediction method to reduce variability of predictions. In particular, they tend to average out not only the variability present between trees (intra-forest variability), but also the variability due to the missing data by fitting a forest for each of the D imputed datasets (between-forest variability). Our theoretical derivation in Section 3 confirm the high potential of MI with ensembles to give accurate predictions. In our empirical investigation (Sections 4 and 5) we examine to what extent these procedures can indeed outperform the other alternatives in practical settings.

We also consider an alternative to multiple imputation, as introduced in [18], that also aims to take into account the variability due to missing data. Since this procedure showed good results in [18], we investigate its performance in our study. The technique first constructs B bootstrap samples from the original incomplete sample. Next, each of these bootstrap samples is imputed once. Although only a single imputation is applied on each bootstrap sample, we end up with B imputed bootstrap samples which may reproduce the variability of the imputation model. In [18] Gaussian, Logistic or Poisson regression is used to generate imputations, but we adapted the procedure by using MIST to impute the bootstrap samples (which thus yields MIST imputed bootstrap samples). This implies that no initial imputation is needed in contrast to the original procedure. RF is then applied on each of the imputed bootstrap samples, resulting in an ensemble of B forests. Finally, the results of all forests are averaged to obtain the final predictions. Similar to the previous strategy both intra-forest variability and between-forest variability is averaged out so that both sampling variability and missing data variability might be taken into account.

3 Theoretical properties

The derivations in this section form a basis to theoretically compare the properties of the methods analyzed in this study. Let us denote by $\varphi_{\mathcal{L}_{\text{miss}},\phi}(\mathbf{x})$ a single tree predictor at $\mathrm{X} = \mathbf{x}$ after imputation of missing values in the training set by a single random draw from their predictive distribution. Here, $\mathcal{L}_{\text{miss}}$ denotes the missing part of the training set and ϕ the single imputation on those data by a given imputation method. For a regression problem, consider the expected generalization error

at $\mathrm{X} = \mathbf{x}$ according to the squared error loss function:

$$\mathrm{E}_{\mathcal{L}}\{\mathrm{Err}\left\{\varphi_{\mathcal{L}_{\mathrm{miss}},\phi}\left(\mathbf{x}\right)\right\}\} = \mathrm{E}_{\mathcal{L}}\{\mathrm{E}_{\mathrm{Y}|\mathrm{X}=\mathbf{x}}\{(Y - \varphi_{\mathcal{L}_{\mathrm{miss}},\phi}\left(\mathbf{x}\right))^2\}\}, \tag{4}$$

where $\mathcal{L}$ denotes the random training set. By rewriting the above expression with respect to the optimal Bayes model φ_{B}, it can be shown that in general the expected generalization error for the prediction at $\mathrm{X} = \mathbf{x}$ additively decomposes into a bias, a variance and a noise component as follows:

$$\begin{aligned}\mathrm{E}_{\mathcal{L}}\{\mathrm{Err}\left\{\varphi_{\mathcal{L}_{\mathrm{miss}},\phi}\left(\mathbf{x}\right)\right\}\} &= (\varphi_{\mathrm{B}}(\mathbf{x}) - \mathrm{E}_{\mathcal{L}}\{\varphi_{\mathcal{L}_{\mathrm{miss}},\phi}(\mathbf{x})\})^2 + \mathrm{E}_{\mathcal{L}}\{(\varphi_{\mathcal{L}_{\mathrm{miss}},\phi}(\mathbf{x}) - \mathrm{E}_{\mathcal{L}}\{\varphi_{\mathcal{L}_{\mathrm{miss}},\phi}(\mathbf{x})\})^2\} \\ &\quad + \mathrm{Err}(\varphi_{\mathrm{B}}(\mathbf{x})) \\ &= \mathrm{bias}^2(\varphi_{\mathcal{L}_{\mathrm{miss}},\phi}(\mathbf{x})) + \mathrm{var}(\varphi_{\mathcal{L}_{\mathrm{miss}},\phi}(\mathbf{x})) + \mathrm{Err}(\varphi_{\mathrm{B}}(\mathbf{x}))\end{aligned} \tag{5}$$

The bias term measures the difference between the average prediction over all possible random training sets and the prediction of the optimal Bayes model. The variance term measures the variability of the predictions generated by $\varphi_{\mathcal{L}_{\mathrm{miss}},\phi}(\mathbf{x})$. Lastly, the third term, $\mathrm{Err}(\varphi_{\mathrm{B}}(\mathbf{x}))$, represents the irreducible error or noise in the data. It is independent of both the prediction method and the training set. This bias-variance decomposition of the expected generalization error was first introduced in [14].

For classification problems a similar decomposition is more difficult to obtain in general. However, several proposals can be found in the literature for the expected generalization error based on the zero-one loss function that give a similar insight into the nature of misclassification error (see e.g.[4, 10, 27, 42]). Moreover, *soft voting*, i.e. averaging class probability estimates and then predicting the most likely class, provide an easy framework to study the generalization error of classification methods by just plugging averaged estimates into (5). This approach yields nearly identical results as *majority voting* [3].

First, we review the results showing when ensemble learning is advantageous in comparison to single model learning in regression. We then adapt these results to show the theoretical advantage of multiple imputation regression trees over single imputation regression trees, given an incomplete training set $\mathcal{L}_{\mathrm{miss}}$. Finally, we extend our results to discuss the theoretical benefit of MI combined with ensembles of trees with respect to SI with an ensemble, MI with a single tree and SI with a single tree.

Ensemble learning

[27] provided theoretical derivations using the bias-variance decomposition to show the superior prediction results of an ensemble of randomized models compared to its single counterpart, given complete training sets. Specifically, let $\mu_{\mathcal{L},\theta}$ denote the expectation of a single randomized predictor $\varphi_{\mathcal{L},\theta}(\mathbf{x})$ (e.g. CART) with randomization parameter θ. θ is considered to be a random variable inducing randomness between the models in an ensemble. Further, let $\sigma^2_{\mathcal{L},\theta}$ denote the variance of such predictor. Now, consider an ensemble of T randomized models (e.g. a forest) $\psi_{\mathcal{L},\theta_1,\cdots,\theta_T}(\mathbf{x}) = \frac{1}{T}\sum_{i=1}^{T}\varphi_{\mathcal{L},\theta_i}(\mathbf{x})$ with $\theta_1,\cdots,\theta_T$ i.i.d. random variables. [27] shows that such an ensemble keeps the same bias as its single model counterpart, but is able to decrease its variability depending on the size of the ensemble T and the correlation $\rho(\mathbf{x})$ between the models in the ensemble. Indeed, we have that

$$\mathrm{E}_{\mathcal{L},\theta_1,\cdots,\theta_T}\{\psi_{\mathcal{L},\theta_1,\cdots,\theta_T}(\mathbf{x})\} = \mu_{\mathcal{L},\theta},$$

and thus it follows that

$$\mathrm{bias}^2(\psi_{\mathcal{L},\theta_1,\cdots,\theta_T}(\mathbf{x})) = (\varphi_{\mathrm{B}}(\mathbf{x}) - \mu_{\mathcal{L},\theta})^2. \tag{6}$$

Hence, an ensemble of randomized models and its single model counterpart have the same bias.

Therefore, ensemble methods can only reduce prediction error by reducing their variance. For the prediction variance of the ensemble we obtain that (see e.g. [27])

$$\text{var}_{\mathcal{L},\theta}\{\psi_{\mathcal{L},\theta_1,\cdots,\theta_T}(\mathbf{x})\} = \rho(\mathbf{x})\sigma^2_{\mathcal{L},\theta}(\mathbf{x}) + \sigma^2_{\mathcal{L},\theta}(\mathbf{x})\left(\frac{1-\rho(\mathbf{x})}{T}\right) \tag{7}$$

with

$$\rho(\mathbf{x}) = \frac{\text{E}_{\mathcal{L},\theta',\theta''}\{\varphi_{\mathcal{L},\theta'}(\mathbf{x})\varphi_{\mathcal{L},\theta''}(\mathbf{x})\} - \mu^2_{\mathcal{L},\theta}(\mathbf{x})}{\sigma^2_{\mathcal{L},\theta}(\mathbf{x})}. \tag{8}$$

If we can make the variance of the ensemble, $\text{var}_{\mathcal{L},\theta}\{\psi_{\mathcal{L},\theta_1,\cdots,\theta_T}(\mathbf{x})\}$, smaller than the single model variance $\sigma^2_{\mathcal{L},\theta}(\mathbf{x})$, then the ensemble improves the prediction performance. As the ensemble gets large, i.e. $T \to \infty$, the variance of the ensemble predictor reduces to $\rho(\mathbf{x})\sigma^2_{\mathcal{L},\theta}(\mathbf{x})$. Hence, large ensembles decrease prediction error when building more decorrelated trees (i.e. with a larger randomization effect). Moreover, for $\rho(\mathbf{x}) \to 0$ the prediction variance reduces to $\frac{\sigma^2_{\mathcal{L},\theta}(\mathbf{x})}{T}$, which again reduces with increasing size T of the ensemble. Notice that when the predictors show no randomization effect at all, i.e. $\rho(\mathbf{x}) \to 1$, then building an ensemble brings no benefit (because all models in the ensemble yield almost the same prediction).

Multiple imputation (MI) versus single imputation (SI) for a single tree

The above results can be extended to the case when MI is combined with single tree prediction given a missing training set $\mathcal{L}_{\text{miss}}$. We assume that the imputed datasets are all obtained by the same imputation strategy but each make a different random draw from the predictive distribution, yielding the prediction $\psi_{\mathcal{L}_{\text{miss}},\phi_1,\ldots,\phi_D}(\mathbf{x}) = \frac{1}{D}\sum_{j=1}^{D}\varphi_{\mathcal{L}_{\text{miss}},\phi_j}(\mathbf{x})$. We can decompose the prediction error as in (5) and similarly as in [27] the expected value and variance of the multiple imputation prediction can be rewritten as:

$$\text{E}_{\mathcal{L},\phi_1,\cdots,\phi_D}\{\psi_{\mathcal{L}_{\text{miss}},\phi_1,\ldots,\phi_D}(\mathbf{x})\} = \mu_{\mathcal{L},\phi},$$

with $\mu_{\mathcal{L},\phi} = \text{E}_{\mathcal{L},\phi}\{\varphi_{\mathcal{L}_{\text{miss}},\phi}\}$. Hence, the bias does not reduce by considering multiple imputations. Therefore, the only source available to reduce prediction error is again the variance of the predictor:

$$\text{var}_{\mathcal{L},\phi_1,\cdots,\phi_D}\{\psi_{\mathcal{L}_{\text{miss}},\phi_1,\ldots,\phi_D}(\mathbf{x})\} = \rho_B(\mathbf{x})\sigma^2_{\mathcal{L},\phi}(\mathbf{x}) + \sigma^2_{\mathcal{L},\phi}(\mathbf{x})\left(\frac{1-\rho_B(\mathbf{x})}{D}\right), \tag{9}$$

where $\sigma^2_{\mathcal{L},\phi}$ is the prediction variance of a single tree with single imputation, and $\rho_B(\mathbf{x})$ is the correlation of trees corresponding to different imputations of the same dataset, namely:

$$\rho_B(\mathbf{x}) = \frac{\text{E}_{\mathcal{L},\phi',\phi''}\{\varphi_{\mathcal{L},\phi'}(\mathbf{x})\varphi_{\mathcal{L},\phi''}(\mathbf{x})\} - \mu^2_{\mathcal{L},\phi}(\mathbf{x})}{\sigma^2_{\mathcal{L},\phi}(\mathbf{x})} \tag{10}$$

Similar conclusions as before can be obtained now. Multiple imputation improves the performance of single imputation increasingly when the number of imputations D increases and when the correlation $\rho_B(\mathbf{x})$ among prediction models on the different imputed datasets decreases. Note therefore the importance of drawing independent imputations to reduce correlation among the different prediction models.

MI + ensemble methods

Now we discuss when MI combined with an ensemble method yields an improvement in prediction performance. The final prediction in this case can be written as

$$\psi_{\mathcal{L}_{\text{miss}},\Lambda}(\mathbf{x}) = \frac{1}{D}\sum_{d=1}^{D}\frac{1}{T}\sum_{t=1}^{T}\varphi_{\mathcal{L}_{\text{miss}}\theta_{t_d},\phi_d}(\mathbf{x}),$$

where Λ denotes a hyperparameter that includes all random parameters θ_{t_d} for growing trees and all random parameters ϕ_d for random imputations.

We consider again the bias-variance decomposition in (5). As before, we assume that the imputed datasets are all obtained by the same imputation strategy but make a different random draw from the predictive distribution. Moreover, we assume again that the randomization parameters θ are i.i.d. random variables. For the bias we obtain that

$$\mathrm{E}_{\mathcal{L},\Lambda}\{\psi_{\mathcal{L}_{\text{miss}},\Lambda}(\mathbf{x})\} = \mathrm{E}_{\mathcal{L},\theta,\phi}\{\varphi_{\mathcal{L}_{\text{miss}},\theta,\phi}\} = \mu_{\mathcal{L},\theta,\phi}.$$

Therefore bias remains the same as when a single predictor with single imputation is used. The component that we address to reduce prediction error is therefore again the variance. We now derive the prediction variance for MI with ensembles.

$$\begin{aligned}
\mathrm{var}_{\mathcal{L},\Lambda}\{\psi_{\mathcal{L}_{\text{miss}},\Lambda}(\mathbf{x})\} &= \mathrm{var}_{\mathcal{L},\Lambda}\{\frac{1}{D}\sum_{d=1}^{D}\frac{1}{T}\sum_{t=1}^{T}\varphi_{\mathcal{L}_{\text{miss}},\theta_{t_d},\phi_d}(\mathbf{x})\} \\
&= \frac{1}{D^2}\frac{1}{T^2}\left[\mathrm{E}_{\mathcal{L},\Lambda}\{(\sum_{d=1}^{D}\sum_{t=1}^{T}\varphi_{\mathcal{L}_{\text{miss}},\theta_{t_d},\phi_d}(\mathbf{x}))^2\} - \mathrm{E}_{\mathcal{L},\Lambda}\{\sum_{d=1}^{D}\sum_{t=1}^{T}\varphi_{\mathcal{L}_{\text{miss}},\theta_{t_d},\phi_d}(\mathbf{x})\}^2\right] \\
&= \frac{1}{D^2}\frac{1}{T^2}\left[\mathrm{E}_{\mathcal{L},\Lambda}\{\sum_{d,e}\sum_{t_d,u_e}\varphi_{\mathcal{L}_{\text{miss}},\theta_{t_d},\phi_d}(\mathbf{x})\varphi_{\mathcal{L}_{\text{miss}},\theta_{u_e},\phi_e}(\mathbf{x})\} - (TD\mu_{\mathcal{L},\theta,\phi}(\mathbf{x}))^2\right] \\
&= \frac{1}{D^2}\frac{1}{T^2}\left[\sum_{d,e}\mathrm{E}_{\mathcal{L},\theta,\phi_d,\phi_e}\{\sum_{t_d,u_e}\varphi_{\mathcal{L}_{\text{miss}},\theta_{t_d},\phi_d}(\mathbf{x})\varphi_{\mathcal{L}_{\text{miss}},\theta_{u_e},\phi_e}(\mathbf{x})\} - T^2D^2\mu^2_{\mathcal{L},\theta,\phi}(\mathbf{x})\right] \\
&= \frac{1}{D^2}\frac{1}{T^2}\Bigg[D\Big(T\,\mathrm{E}_{\mathcal{L},\theta,\phi}\{\varphi_{\mathcal{L}_{\text{miss}},\theta,\phi}(\mathbf{x})^2\} + (T^2-T)\,\mathrm{E}_{\mathcal{L},\theta',\theta'',\phi}\{\varphi_{\mathcal{L}_{\text{miss}},\theta',\phi}(\mathbf{x})\varphi_{\mathcal{L}_{\text{miss}},\theta'',\phi}(\mathbf{x})\}\Big) \\
&\quad + (D^2-D)\Big(T^2\,\mathrm{E}_{\mathcal{L},\theta',\theta'',\phi',\phi''}\{\varphi_{\mathcal{L}_{\text{miss}},\theta',\phi'}(\mathbf{x})\varphi_{\mathcal{L}_{\text{miss}},\theta'',\phi''}(\mathbf{x})\}\Big) - T^2D^2\mu^2_{\mathcal{L},\theta,\phi}(\mathbf{x})\Bigg] \\
&= \frac{1}{D^2}\frac{1}{T^2}\Bigg[D\Big(T(\sigma^2_{\mathcal{L}_{\text{miss}},\theta,\phi}(\mathbf{x}) + \mu^2_{\mathcal{L},\theta,\phi}(\mathbf{x})) + (T^2-T)(\rho_W(\mathbf{x})\sigma^2_{\mathcal{L}_{\text{miss}},\theta,\phi}(\mathbf{x}) + \mu^2_{\mathcal{L},\theta,\phi}(\mathbf{x}))\Big) \\
&\quad + (D^2-D)\Big(T^2(\rho_B(\mathbf{x})\sigma^2_{\mathcal{L}_{\text{miss}},\theta,\phi}(\mathbf{x}) + \mu^2_{\mathcal{L},\theta,\phi}(\mathbf{x}))\Big) - T^2D^2\mu^2_{\mathcal{L},\theta,\phi}(\mathbf{x})\Bigg] \\
&= \frac{\rho_W(\mathbf{x})\sigma^2_{\mathcal{L}_{\text{miss}},\theta,\phi}(\mathbf{x})}{D} + \sigma^2_{\mathcal{L}_{\text{miss}},\theta,\phi}(\mathbf{x})\left(\frac{1-\rho_W(\mathbf{x})}{D\cdot T}\right) + \rho_B(\mathbf{x})\sigma^2_{\mathcal{L}_{\text{miss}},\theta,\phi}(\mathbf{x})\left(1-\frac{1}{D}\right)
\end{aligned} \tag{11}$$

where $\rho_W(\mathbf{x})$ is the correlation of trees fitted on the same imputed dataset. More specifically:

$$\rho_W(\mathbf{x}) = \frac{\mathrm{E}_{\mathcal{L},\theta',\theta'',\phi}\{\varphi_{\mathcal{L},\theta',\phi}(\mathbf{x})\varphi_{\mathcal{L},\theta'',\phi}(\mathbf{x})\} - \mu^2_{\mathcal{L},\theta,\phi}(\mathbf{x})}{\sigma^2_{\mathcal{L},\theta,\phi}(\mathbf{x})} \tag{12}$$

Note that $\sigma^2_{\mathcal{L}_{\text{miss}},\theta,\phi}$ is the variance of a single tree after single imputation. If we can make the variance in (11) smaller than $\sigma^2_{\mathcal{L}_{\text{miss}},\theta,\phi}$, then the prediction error of MI ensembles will be lower than that of SI with single trees. Remark that the first two terms in (11) are related to the sampling variability of the predictions while the last term is related to the extra variability in the predictions caused by the missing values. From (11) we can also see that if we take the number of imputations D large enough, having a low correlation $\rho_W(\mathbf{x})$ among the trees in each ensemble is not a necessary condition to decrease prediction error. It then suffices to decrease the correlations among imputations $\rho_B(\mathbf{x})$. This is in correspondence with our previous findings for MI + single trees.

The variance of MI with ensembles can be linked to the variance of MI with single trees in (9) by rewriting (11) as follows.

$$\begin{aligned}\mathrm{var}_{\mathcal{L},\Lambda}\{\psi_{\mathcal{L}_{\text{miss}},\Lambda}(\mathbf{x})\} = \rho_B(\mathbf{x})\sigma^2_{\mathcal{L}_{\text{miss}},\theta,\phi}(\mathbf{x}) + \frac{\sigma^2_{\mathcal{L}_{\text{miss}},\theta,\phi}(\mathbf{x})}{T}\left(\frac{1-\rho_B(\mathbf{x})T}{D}\right) \\ + \frac{\rho_W(\mathbf{x})\sigma^2_{\mathcal{L}_{\text{miss}},\theta,\phi}(\mathbf{x})}{D} - \frac{\rho_W(\mathbf{x})\sigma^2_{\mathcal{L}_{\text{miss}},\theta,\phi}(\mathbf{x})}{D\cdot T}\end{aligned} \tag{13}$$

Comparing the expression in (9) to (13) reveals that a lower prediction variance for MI with ensembles can be achieved by fitting a large number of decorrelated trees on a large number of decorrelated imputed datasets. While MI with single trees only reduces variability in the predictions due to missing data, MI with ensembles also reduces the sampling variability of the predictions.

A similar comparison can be carried out for SI followed by an ensemble which yields the predictor $\psi_{\mathcal{L}_{\text{miss}},\theta_1,\ldots,\theta_T,\phi}(\mathbf{x}) = \frac{1}{T}\sum_{t=1}^{T}\varphi_{\mathcal{L}_{\text{miss}},\theta_t,\phi}(\mathbf{x})$. This predictor again has the same bias as the SI + single tree predictor. Moreover, the prediction variance of the SI with ensemble predictor becomes:

$$\mathrm{var}_{\mathcal{L},\theta_1,\cdots,\theta_T,\phi}\{\psi_{\mathcal{L}_{\text{miss}},\theta_1,\ldots,\theta_T,\phi}(\mathbf{x})\} = \rho_W(\mathbf{x})\sigma^2_{\mathcal{L},\theta,\phi}(\mathbf{x}) + \sigma^2_{\mathcal{L},\theta,\phi}(\mathbf{x})\left(\frac{1-\rho_W(\mathbf{x})}{T}\right), \tag{14}$$

By comparing (14) to (11) it is immediately clear that the performance of SI with ensemble can be improved by imputing the training data several times with decorrelated imputations (i.e. with MI ensembles).

Finally, we conclude that MI with ensembles gives superior results to SI with an ensemble, MI with a single tree and SI with a single tree. Surrogate splits can be considered as a special case of single imputation [30], so we can expect that MI with ensembles will also yield better performance than surrogates. Therefore, theoretically MICE + CondRF forms an ideal combination. Indeed, by construction MICE attempts to make independent draws for the imputations while at the same time CondRF attempts to grow decorrelated trees. Using CondRF also helps to improve the whole technique by reducing bias.

4 Simulation study

In order to compare the use of surrogates versus imputation, and more in general the predictive performance of the 26 methods considered (see Table 1), empirical studies similar to those in [13, 15, 32] were performed. Next to comparing the empirical performance with the theoretical conclusions, it is of interest to compare our findings with the results in these previous studies, especially with those in the most recent work [15]. Hence, all four real-life datasets without missing data selected in [15] have also been used in our studies. They comprise datasets available in R [31] and datasets from the UCI Machine Learning Repository [1]. Two of these datasets concern classification and the other two are regression problems. In addition, we also considered a simulated regression dataset where the response follows the data generating model (DGM) of the simulation study in [8]. An overview of the total number of observations and predictors in each dataset can be found in Table 2. We now give a short summary of these datasets.

- The *Haberman's Survival Dataset* contains 306 cases from a study conducted on patients who had undergone surgery for breast cancer. It can be obtained from the UCI Machine Learning Repository [1]. We aim to predict the 5-year survival status of a patient based on the three available predictor variables.
- The *Statlog (Heart) Disease Dataset* was collected from 270 patients at four different hospitals. It is provided by the UCI Machine Learning Repository [1]. Our objective is to predict the presence of heart disease based on 13 clinical measurements of the patients.
- The *Swiss Fertility and Socioeconomic Indicators Dataset* was collected at 47 French-speaking provinces of Switzerland around 1888. It is provided by R [31] and is used to predict a standardized fertility measure from a set of 5 socio-economic indicators.
- The *Infant Birth Weight Dataset* was gathered from 189 newborns at the Baystate Medical Center, Springfield, Mass, during the year 1986. It is available in the R package MASS and is used to predict the baby's birth weight in grams from 8 risk factors.
- A large regression dataset was generated in order to assess our research questions in a possibly more complex context that might be present in real-life situations. In particular, a dataset with 500 observations and ten continuous predictors was created. Predictors were generated from linear models in a way that complexities such as circular dependence, multicollinearity and interactions may be present. To generate the response variable, the DGM specified in the simulation study of [8] was used with the same parameter values. The noise to total variability ratio was kept lower than 10% throughout the generation of the variables. Detailed information on the DGM for this dataset can be found in the supplementary material.

Table 2: Number of observations and predictors listed for each dataset used in this study

Dataset	Obs.	Var.
Survival	306	3
Heart	270	13
Fertility	47	5
Birthweight	189	8
Simulated	500	10

To make our findings comparable to those in [15], missing values were introduced in a similar way as in their paper, although only in the training data. We only considered complete test cases for evaluation purposes, to avoid an extra source of variability in the performance measures. In accordance with the recommendation in [15] to investigate a wider range of patterns we did not only introduce missing values completely at random (MCAR) but also at random (MAR) and not at random (MNAR). We now discuss the missing data mechanisms used in our study in more detail.

Real-life datasets

The following steps were used to introduce missing data in the real datasets according to the different missing data mechanisms and schemes:

1. Randomly split dataset: 80% training set, 20% test set.
2. Fix the fraction of missing data for each variable with missing values as $\eta = 10\%$, 20%, 30%, or 40%.
3. Insert missing data in the training set according to one of the following missing data mechanisms and schemes.
   - *Under MCAR mechanism*:
     First scheme: Randomly induce missing data in ALL (p) variables. In each variable a fraction η of missing values is inserted at random.
     Second scheme: Induce missing data in ONE THIRD of all variables ($p/3$) chosen at random. In each of these variables a fraction η of missing values is inserted at random.
   - *Under MAR mechanism*:
     First scheme: Randomly choose one "determining variable" x_{det} to induce missing data in the remaining $p-1$ variables. In each variable a fraction η of missing values is inserted. To this end, the value of the determining variable is transformed into a probability by a logistic function. A missingness indicator is then generated from a Bernoulli distribution with this probability.
     Second scheme: Induce missing data in ONE THIRD of all variables ($p/3$) chosen at random. In each of these variables a fraction η of missing values is inserted. The remaining two thirds of variables now form the "determining variables". The values of these determining variables are transformed into a probability by a logistic function. The missingness indicator is then generated from a Bernoulli distribution with this probability.
   - *Under MNAR mechanism*:
     First scheme: Induce missing data in ALL (p) variables. In each variable a fraction η of missing values is inserted based on its upper or lower η quantile (we change this from dataset to dataset), i.e. in every variable a missing status is given to observations that are above (or below) its upper (or lower) η quantile.
     Second scheme: Induce missing data in ONE THIRD of all variables ($p/3$) chosen at random. In each variable a fraction η of missing values is inserted based on its upper or lower η quantile (we change this from dataset to dataset), i.e. in every variable a missing status is given to observations that are above (or below) its upper (or lower) η quantile.

Note that in the first scheme of MCAR and MNAR it can happen that an observation has missing values for all the predictor variables. Such observations were removed from the dataset because they cause problems for several imputation methods.

Simulated dataset

A similar design was used for this dataset. More specifically, these are the steps taken for the introduction of missing data in our simulated data:

1. Randomly split dataset: 80% training set, 20% test set.
2. Fix the fraction of missing data for each variable with missing values as $\eta = 10\%$, 20%, 30%, or 40%.
3. Insert missing data in the training set according to the different mechanisms and schemes.

   - *Under MCAR mechanism*:
     First scheme: Randomly induce missing data in the first 8 variables $(x_1, x_2, \ldots, x_8)$. In each variable a fraction η of missing values is inserted at random.
     Second scheme: Randomly induce missing data in ONE THIRD of all variables $(p/3)$ chosen at random. In each variable a fraction η of missing values is inserted at random.
   - *Under MAR mechanism*:
     First scheme: Use x_9 and x_{10} as potential "determining variables" to induce missing data in $(x_1, x_2, \ldots, x_8)$. In each variable a fraction η of missing values is inserted by randomly selecting one of the following three strategies:
     - insert missing values based on the upper η quantile of one randomly chosen "determining variable" among x_9 and x_{10}, i.e. in every variable a missing status is given to observations that correspond with those of the chosen "determining variable" that are above this upper η quantile.
     - insert missing values as in the previous strategy but now using the lower η quantile.
     - use both x_9 and x_{10} as determining variables and transform their values into a probability by a logistic function. A missingness indicator is then generated from a Bernoulli distribution with this probability.

     Second scheme: Induce missing data in ONE THIRD of all variables $(p/3)$ chosen at random. In each variable a fraction η of missing values is inserted based on the potential "determining variables" x_9 and x_{10} following the same procedure as in the previous scheme.
   - *Under MNAR mechanism*:
     First scheme: Induce missing data in the first 8 variables $(x_1, x_2, \ldots, x_8)$. In each variable a fraction η of missing values is inserted based on its upper η quantile. That is, a missing status is given to observations that are above this upper quantile.
     Second scheme: Induce missing data in ONE THIRD of all variables $(p/3)$ chosen at random. In each variable a fraction η of missing values is inserted based on its upper η quantile. That is, a missing status is given to observations that are above this upper quantile.

General

As in related studies, predictive performance was assessed via the *mean squared prediction error* (MSPE) for regression or its equivalent *misclassification error* (MER) for classification. The procedure to generate datasets with missing values was repeated $1,000$ times for each mechanism and scheme. The mean root MSPE (RMSPE) or the mean MER across these $1,000$ iterations is reported as a final measure of predictive performance. Moreover, a measure for the performance improvement with an imputation strategy compared to surrogate decisions is calculated as in [15]:

$$\text{rel.impr.} = \frac{\text{MSPE}_{\text{Sur.}} - \text{MSPE}_{\text{Imp.}}}{\text{MSPE}_{\text{Sur.}}}. \tag{15}$$

Hence, we report the mean relative improvement to assess the performance of an imputation method compared to surrogates.

All simulations were implemented in the R statistical software [31]. To allow a fair comparison with [15], R function settings in their paper were replicated in our study. An overview of all the settings for the methods used in our empirical studies is given in Table 3. As mentioned earlier, the R package randomForest [25] does not support the use of surrogate decisions. Therefore, no comparison between surrogates and imputation could be made for RF.

Table 3: R function and its corresponding package name, package reference paper and settings for the implementation of each of the methods included in this study

Technique	R function	R package	Reference	R Settings
CART	rpart()	rpart	[41]	maxsurrogate = min(3, variables available)
CondTree	ctree()	party	[20]	maxsurrogate = min(3, variables available)
RF	randomForest()	randomForest	[25]	ntree = 500, mtry = min(5, variables available)
CondRF	cforest()	party	[20]	ntree = 500, mtry = min(5, variables available), maxsurrogate = min(3, variables available)
Bagging	bagging()	ipred	[29]	nbagg= 500, maxsurrogate = min(3, variables available)
CondBagging	cforest()	party	[20]	ntree = 500, maxsurrogate = min(3, variables available)
Median/mode	na.roughfix()	randomForest	[25]	none
Prox. matrix	rfImpute()	randomForest	[25]	ntree = 500, mtry = min(5, variables available), iter = 5 [a]
MICE	mice()	mice	[47]	m = 5, defaultMethod = c("norm","logreg","polyreg")
MIST	treeMI()	treeMI	[8]	ITER = 20
kNN	kNNImpute() [b]	imputation	[43]	k=5

[a] Error messages were displayed frequently when running the rfImpute() routine with iter = 5 on datasets with large amounts of values MAR or MNAR. To obtain imputation in these cases, we ran this routine exceptionally with iter = 1 combined with median/mode imputation when no convergence of the Prox. matrix algorithm was attained at some cells.

[b] Since the kNNImpute() routine only allows numeric data as input, techniques with prior kNN imputation could not be included in the comparisons made on datasets with at least one categorical predictor (Heart and Birthweight datasets).

It has to be emphasized that our comparisons were made among 26 techniques with fixed modeling strategies. Issues like estimation of parameters that yield the best tree structure or setting the best possible imputation model for a given imputation strategy are outside the scope of this study. These settings were specified to allow comparability.

As in [15], the recommendations of [23] on the proper publication of imputation methods were followed in this paper. They are outlined in this Section and described in more detail in the supplementary material. This allows researchers in the field to evaluate the impact of these methods in practice by looking at every result and the particular situation(s) in which they hold.

5 Results and Discussion

A summary of mean RMSPE/MER values as the percentage of missing data increases can be found in Figures 1-3 for all datasets analyzed in this study. To make the plots more informative, we decided to remove all methods with overall low performance. We only show the results for schemes with all variables containing missing values (first schemes). The performance of all methods when a random third of the variables contains missing values is quite stable and resembles that of 10% missingness in all variables. Note that in the plots we have used the same point characters for techniques based on the same tree prediction method. Different line types and colors (gray scale) correspond to the different missing data treatments. In addition to these graphical results, a general summary of mean relative improvement values can be found in Table 4 and 5. More extensive numerical reports of mean MSPE/MER values and mean relative improvement values can be found in the supplementary material.

5.1 Comparison of techniques

The lower lines in Figures 1-3 mostly correspond to ensemble methods. This confirms the theoretical result in Section 3 that the usage of ensemble methods is advisable when the goal is prediction. These methods benefit from their ensemble nature to average out sampling variability. The same result was empirically obtained by [15] and corresponds to what has been broadly shown by several authors, e.g. [3, 7]. Throughout our simulations CondRF and RF methods as well as CondBagging performed in general superior to single tree methods. Among these ensemble methods, one especially finds that the combinations MICE/MIST + CondRF, MICE/MIST + RF, Prox. matrix + RF and CondBagging alternatively beat each other throughout the datasets and scenarios analyzed.

For small amounts of missing data, the plots in Figures 1-3 show that CondRF/CondBagging with surrogates or RF/CondRF with a previous single imputation suffices in general to obtain good prediction results. Therefore, we do not need to make more intensive multiple imputation computations to obtain satisfactory predictions under this scenario. In particular, CondRF with surrogates (dotted lines with triangle point-down symbols) performs as well as other CondRF combinations in three out of the four real-life datasets: Survival, Heart and Birthweight. Similarly, a SI + RF strategy is sufficient to obtain competitive prediction results in the Fertility, Heart and simulated datasets. For instance, Prox. matrix + RF (long-dashed lines with triangle point-up symbols) performs very well for these datasets.

When the amount of missing values is large under the MCAR or MAR patterns MICE/MIST + CondRF/RF methods perform well throughout the datasets analyzed. MICE/MIST + CondRF are shown in Figures 1-3 with the triangle point-down joined with solid thick lines for MICE as opposed to solid thin lines for MIST. Both methods show competitive performance in comparison to the other techniques in three real-life datasets: Survival, Heart and Birthweight. RF methods (portrayed by the triangle point-up) achieve the first place in the Fertility dataset, with all missing data methods performing equally well, while in the simulated dataset Prox. matrix + RF performs best; in both cases with a clear difference over the other techniques.

When the amount of missingness becomes large under the MNAR pattern, simulations show that MICE/MIST + CondRF again produces satisfactory results in general. In most instances of the real-life datasets they are at least competitive to the other methods. On the other hand, in our studies the performance of RF methods systematically deteriorates relative to the other methods. This is particularly the case for all RF methods in the Survival and Fertility datasets, for RF combined with SI in the Heart and Birthweight datasets and for MICE + RF in the simulated dataset. In Figure 2 one can note that for the Fertility dataset RF methods goes down from being the best techniques under MCAR and MAR mechanisms to become incredibly the worse techniques at large amounts of data MNAR. Likewise, the plots for the Survival, Birthweight and simulated datasets report more robustness in terms of predictive performance for CondRF methods compared to RF methods under this complex missing data scenario. This tendency of RF methods was further observed in other simulation studies whose results are not shown here. Most likely, the difference in performance between CondRF and RF methods resides on the different strategy that is used to select a splitting covariate in each region. The CART procedure in RF might bias the selection of splitting variables while the conditional trees in CondRF aim to prevent this. As a result, CondRF methods can be more successful in extracting valuable information from the (imputed) predictor variables than RF methods, especially in situations of high uncertainty caused by a complex missing data structure.

Interestingly, CondBagging with surrogate decisions (dotted lines with + symbol) also yielded quite competitive results for all datasets and different scenarios analyzed. Moreover, it is also computationally much faster than MICE/MIST + CondRF/RF (see subsection 5.4), which is an extra advantage. Bagging, however, always showed worse performance than CondBagging, even for small amounts of missing data.

For SI, it turns out that imputation by the Prox. matrix performs in general comparable to kNN imputation (results shown in the supplementary material). The new method of MIST imputed bootstrap samples + RF (in gray solid line in Figures 1-3) shows good results in general in comparison to techniques that combine a single tree with surrogates or to techniques that combine RF with single imputation. This is in line with the results in [18]. However, when compared to CondRF procedures or RF combined with MICE or MIST, it turns out that it has a comparable or slightly worse performance, but never yields a real improvement.

A comparison between the multiple imputation methods MICE (in solid thick lines) and MIST (in solid thin lines) in Figures 1-3 reveals that they mostly yield quite similar prediction results, with a slight advantage for MICE in the real datasets. Hence, in most cases the extra flexibility by using trees in MIST does not lead to better imputations, due to the higher variability of this procedure. However, the high flexibility of MIST may be useful to capture complicated structures in complex datasets. This is the case for the simulated dataset where MIST yielded better results in comparison to MICE. [11] showed similar results for MIST with complex simulated datasets involving different types of interactions, considering both categorical and continuous responses.

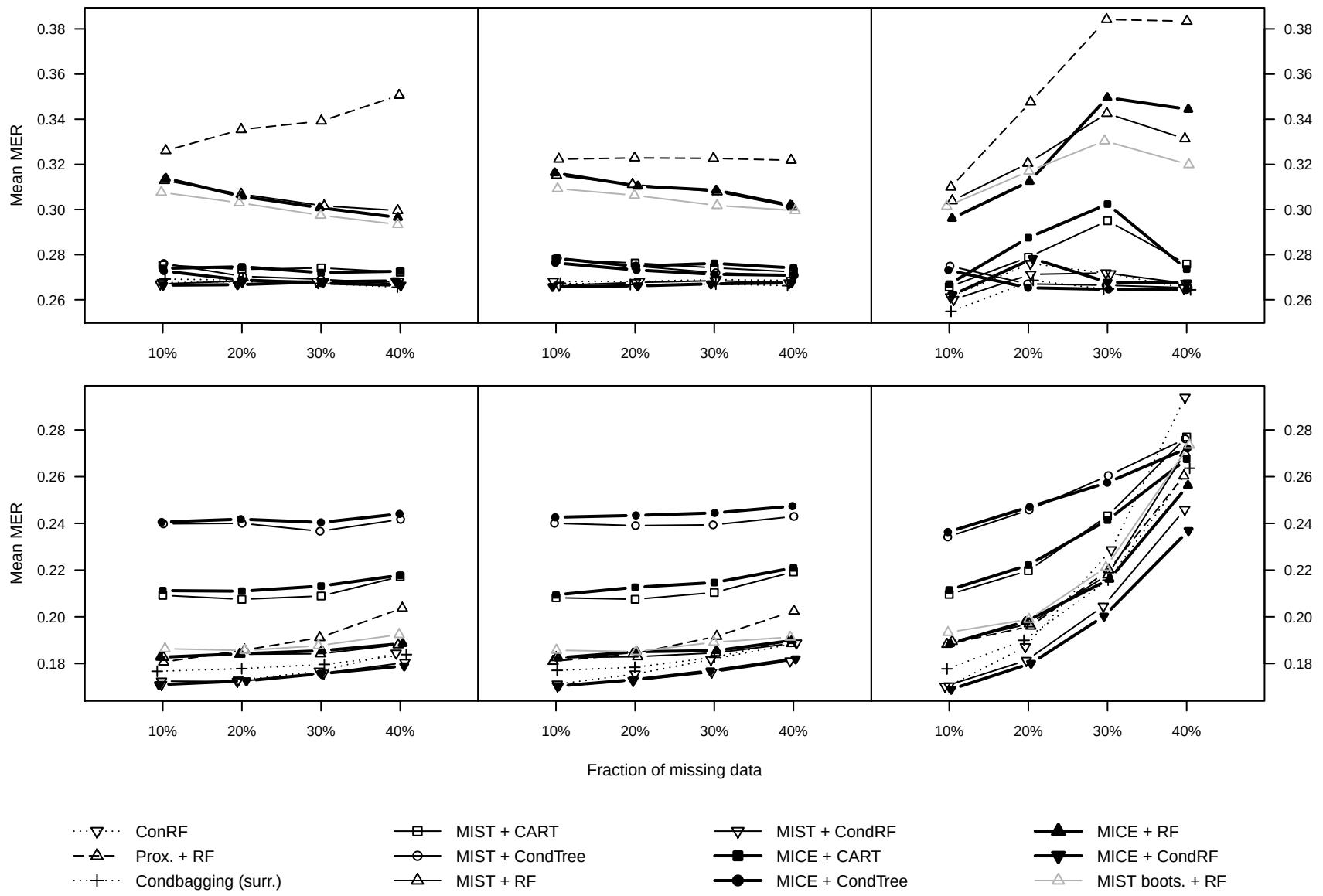

Figure 1: Mean MER results for the Survival data (top row) and Heart disease data (bottom row). Results are shown for data MCAR (left panel), MAR (middle panel) and MNAR (right panel).

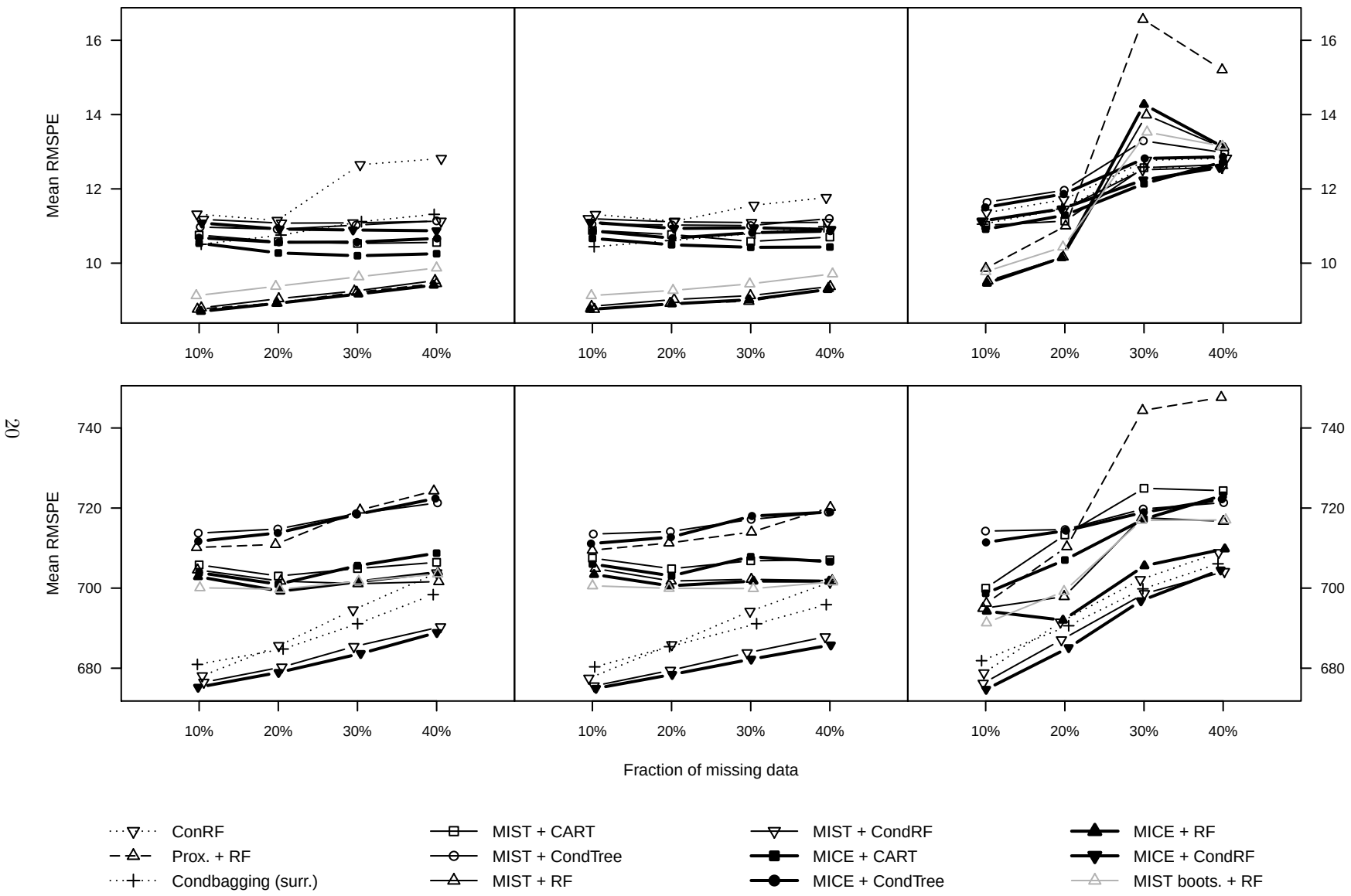

Figure 2: Mean RMSPE results for the Fertility data (top row) and Birthweight data (bottom row). The values for the Birthweight dataset have been divided by 10^5. Results are shown for data MCAR (left panel), MAR (middle panel) and MNAR (right panel).

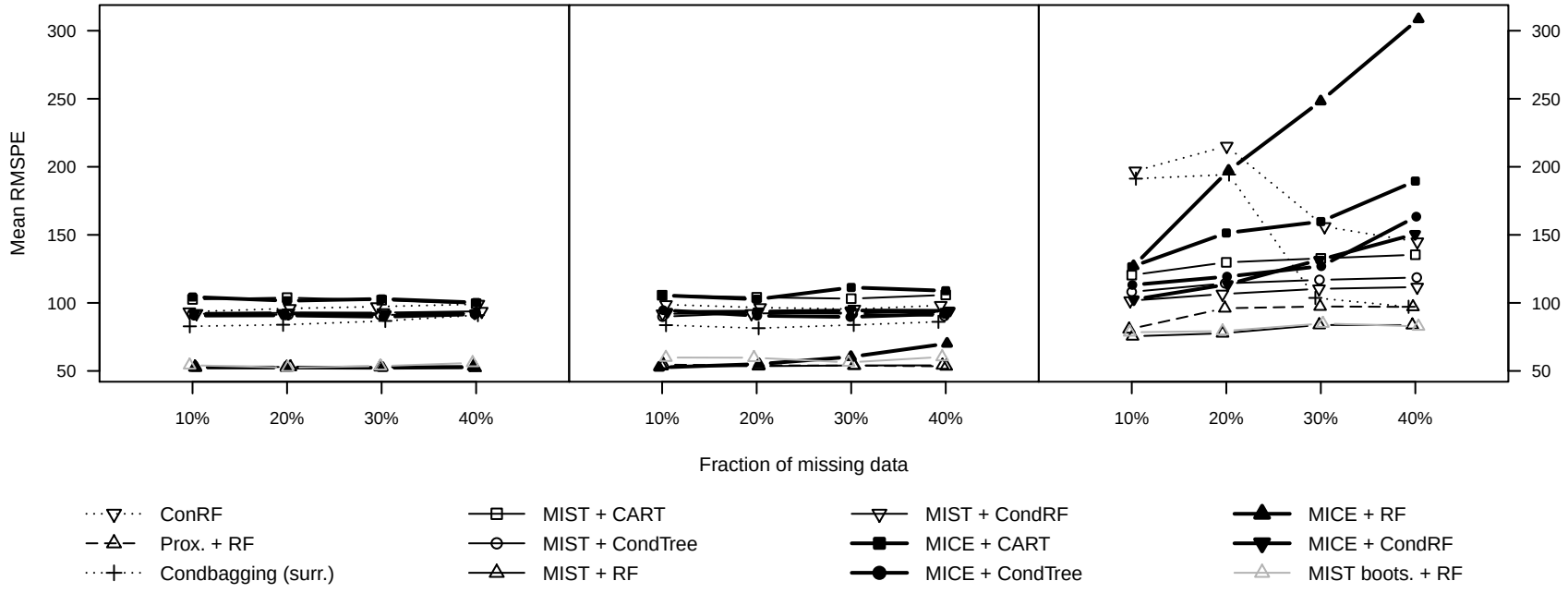

Figure 3: Mean RMSPE results for the simulated data. Results are shown for data MCAR (left), MAR (middle) and MNAR (right).

5.2 Effects of sample size and dimension

We carried out experiments to investigate the effect of different sample sizes and dimensions on the performance of the techniques under investigation. We used the design of the simulated dataset and the MNAR scenario. This is the most complex scenario and exhibits the largest differences across the different percentages of missing data. Performance patterns for MCAR and MAR mechanisms were quite similar, but with lower error rates than for MNAR values. First, the sample size was extended to 750, 1000 and 2000 observations while the dimension remained fixed at 10. Figure 4 shows the results. When the sample size increases, the prediction errors of the methods change very little with a slight tendency to decrease for some methods. The general performance pattern of the methods is almost not changed. Thus, Prox. matrix + RF, MIST + RF and MIST imputed bootstrap samples + RF keep their good performance when the sample size increases.

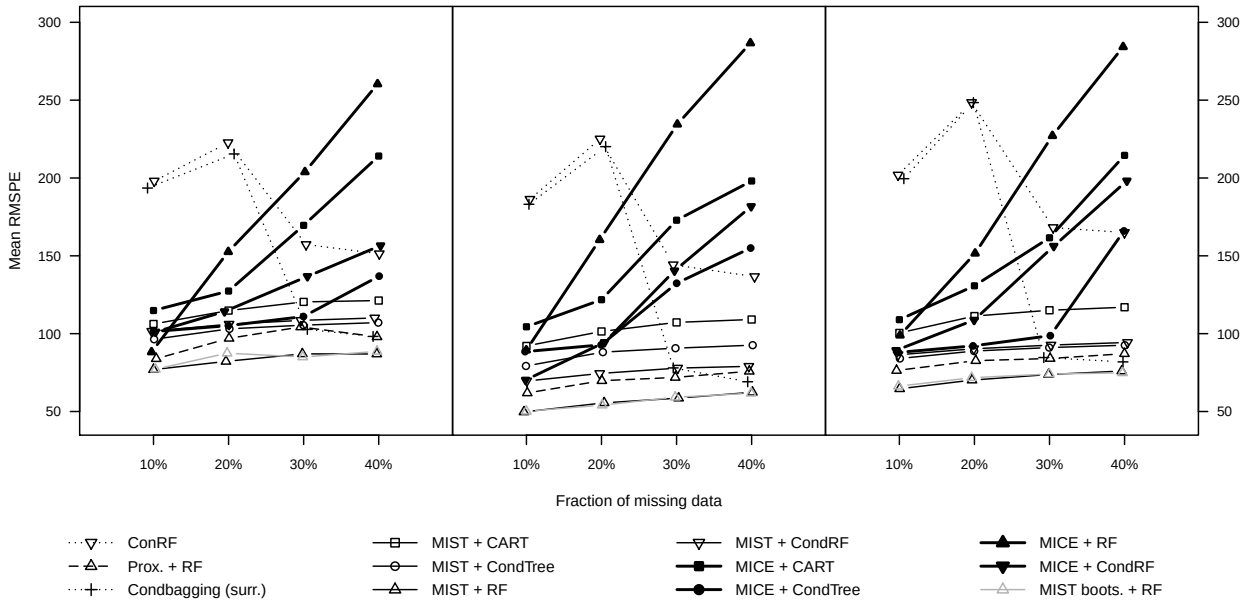

Figure 4: Effect of sample size on the prediction performance for simulated datasets under the MNAR mechanism. Results are shown for sample sizes with 750 observations (left panel), 1000 observations (middle panel) and 2000 observations (right panel) with dimension fixed at $p = 10$.

Secondly, we kept the sample size fixed at 500, but increased the dimension to 15, 20 and 50 continuous predictors, by adding noise predictors to our simulated dataset. Missing data were also generated for these noise predictors. These results are shown in Figure 5. When the dimension grows, most prediction errors slightly increase when compared to the original simulated dataset. In contrast, CART or CondTree combined with multiple imputation by MICE yield better prediction errors in higher dimensions, which become comparable to those of their counterparts using MIST. While MIST performed clearly better than MICE in the original 10 dimensional dataset, the difference between both approaches becomes smaller as the dimension grows. Although this effect seems small for the range of dimension we consider, intuitively this effect could be expected.

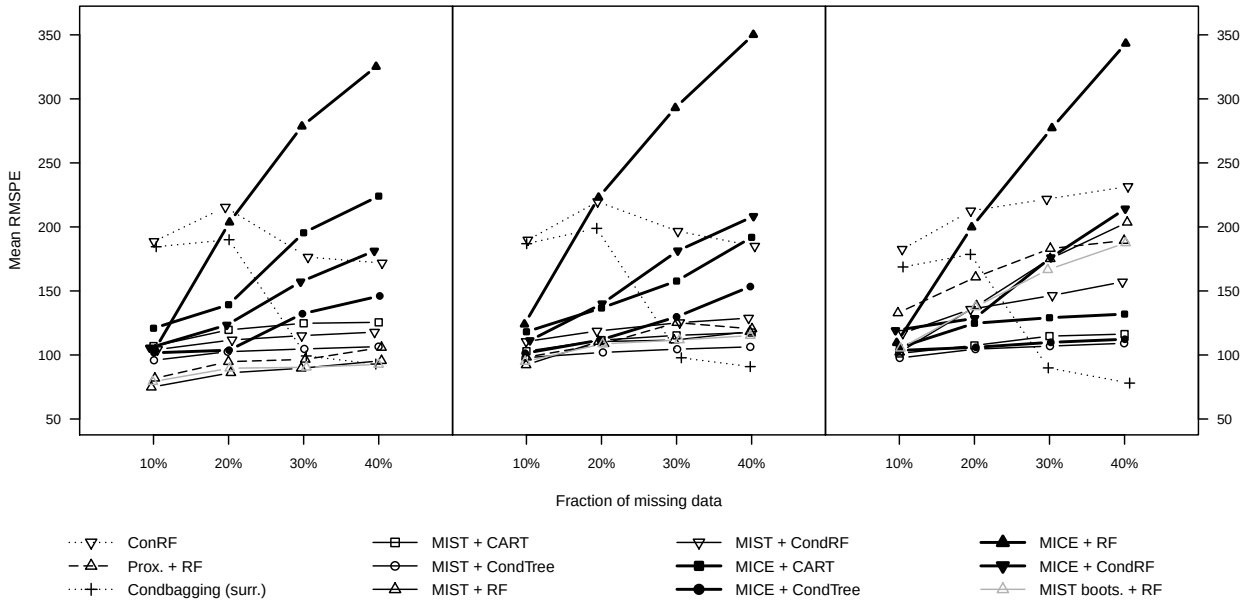

Figure 5: Effect of dimension on the prediction performance for simulated datasets under the MNAR mechanism. Results are shown for dimensions with 15 (left panel), 20 (middle panel) and 50 continuous predictors (right panel) with sample size fixed at $N = 500$.

MICE uses more rigid linear models to make imputations while MIST uses flexible models to make its imputations (i.e. CART models). The linear models in MICE are more biased but less variable than the models in MIST. In higher dimensions with a lot of noise variables, the lower variance of the linear models helps MICE to introduce stability in the imputations and therefore stability in the predictions. Note also that CondBagging and CondRF (both with surrogates) keep their performance stable as the dimension grows (with better performance for CondBagging) in contrast to some methods based on imputation which show a clear increase in their prediction error (e.g. MIST + RF/CondRF). This may imply that the variability reduction by averaging in MI is exceeded by the extra noise introduced in the imputation process, due to the many noise predictors.

We also investigated the combined effect of sample size and dimension by generating datasets of size 1000 in 50 dimensions. The effect on prediction error (results not shown) was less pronounced than for sample size 500. Therefore, CondBagging can be advised for large dimensional datasets, that likely contain noisy variables.

5.3 Initial imputation versus surrogates

We first compare SI to the use of surrogate decisions. Table 4 shows ranges of mean relative improvement values over all techniques with SI and all missing data fractions. These ranges are specified for the three missing data mechanisms: MCAR, MAR and MNAR. In general there is no clear improvement by SI. As can be seen in Table 4, sometimes large improvements occur but they are not regular throughout the analysis. For instance, for the Heart and Survival datasets no SI method yields a clear improvement, except at a few instances with MNAR data (e.g. Prox. matrix + CondRF with 31%). For the simulated dataset, single imputation by Prox. matrix or

kNN sometimes yields an improvement for moderate to large fractions of missing data (e.g. kNN + CondRF with 80%), while in the Birthweight dataset only single imputation by Prox. matrix combined with CondRF slightly improves on surrogates (e.g. 4% under MAR). For the Fertility dataset the largest improvement rates are obtained by kNN imputation (22% for MCAR and 5% for MAR). Overall, there is no guarantee that SI will be superior to surrogates in real-life applications and in fact it can turn out to be much worse as can be seen from the large negative lower bounds in Table 4.

Table 4: Ranges of mean relative improvement for single imputation over all techniques and missing data fractions. Note that the first result line corresponds to the MCAR pattern, the second to the MAR and the third to the MNAR pattern. Only CondRF, CondTree and CART were taken into account for these comparisons.

Fraction of var. miss.	Real-life Datasets				Simulated dataset
	Haberman's Survival	Heart Disease	Swiss Fertility	Birthweight	
1/1	-6% to 1%	-17% to 0%	-11% to 22%	-11% to 3%	-46% to 44%
	-4% to 1%	-11% to -1%	-14% to 5%	-9% to 4%	-14% to 29%
	-20% to 4%	-15% to 31%	-53% to 4%	-12% to 1%	-63% to 80%
1/3	-2% to 0%	-17% to -1%	-15% to 0%	-5% to 1%	-3% to 2%
	-2% to 0%	-17% to -2%	-15% to 1%	-5% to 1%	-2% to 1%
	-3% to 1%	-14% to 0%	-20% to 1%	-5% to 1%	-12% to 1%

Multiple imputation followed by a single tree method (CART or CondTree) in general performs better than surrogates when having a high fraction of data missing on all features under any pattern. However, as in [15], we emphasize that the comparison between MI and surrogates for single trees is not fair. The reason is that MI combined with single trees already has an ensemble nature as seen in Section 3. Therefore, it is more honest to look at improvement rates by MI when using an ensemble method instead of single trees.

Table 5 contains the mean improvement rates of MI with CondRF with respect to CondRF with surrogates. MI followed by an ensemble method does not yield a distinct benefit over surrogates when the amount of incomplete data is small. This is mostly indicated by the rates on the left extremes of the ranges in Table 5. However, it can yield an improvement when the amount of missingness increases. For instance, for the real datasets MI + CondRF often yields much better results than surrogates at large amounts of data missing in all covariates (1/1) and for any pattern. The latter is mostly shown by the rates on the right extremes of the ranges in Table 5. In the simulated dataset only MIST + CondRF performs clearly better than surrogates, reaching an improvement rate of 80%, while MICE + CondRF performs clearly worse (see Figure 3).

Note that our results for the MCAR pattern differ from those obtained in [15] with real-life datasets originally containing missing values. Authors in [15] concluded that there was no convincing improvement when using MI compared to surrogates. However, there are some differences with our study concerning the modeling of the imputation distributions, as discussed in Section 1, which can explain the difference in results.

5.4 Computational issues

The good and safe performance of MICE/MIST + CondRF techniques comes at a cost in terms of computation time. CondBagging arises as the best alternative when one is interested in making a tradeoff between performance and computational speed. It showed quite good results throughout

Table 5: Ranges of mean relative improvement for multiple imputation with CondRF vs CondRF with surrogates over all missing data fractions. Note that the first result line corresponds to the MCAR pattern, the second to the MAR and the third to the MNAR pattern.

Fraction of var. miss.		Real-life Datasets			Simulated dataset
	Haberman's Survival	Heart Disease	Swiss Fertility	Birthweight	
1/1	-1% to 0%	-2% to 1%	0% to 27%	0% to 4%	5% to 20%
	0% to 1%	0% to 2%	-1% to 11%	1% to 4%	-28% to 7%
	-1% to 1%	-1% to 19%	2% to 8%	1% to 2%	-108% to 80%
1/3	-1% to 0%	-1% to 3%	1% to 3%	0% to 2%	-1% to 1%
	-1% to 0%	-1% to 2%	0% to 4%	0% to 2%	-6% to 1%
	-1% to 0%	-1% to 2%	-1% to 2%	0% to 2%	-52% to 1%

the simulation study and it shows a fairly stable computation time even with increasing amounts of missing data. Plots of predictive performance versus average computation time (in CPU seconds) are shown in Figure 6 for the Birthweight, Heart, Fertility and simulated datasets under the MNAR mechanism. This is the scenario that shows the largest differences in performance and computational cost. However, similar conclusions can be obtained with other scenarios. The different points in the plots indicate the different percentages of missing data introduced (10%, 20%, 30% and 40%). Note that the computation times are expressed in seconds and were obtained on a single Intel i7 CPU (3.4GHz) machine running Windows 7.

From the plots, the relatively fast computation of CondBagging is evident (see dotted lines with + symbol). Compared to MIST + CondRF, it runs at least 10 times faster for the Birthweight, 30 times faster for the Heart, 6 times faster for the Fertility and 10 times faster for the simulated dataset, while both methods show similar performance in many cases. The nice trade-off between performance and computation time for CondBagging may become less important when the practitioner has access to multiple processor machines because the MICE/MIST + CondRF procedures can easily be parallelized.

The plots in Figure 6 also suggest that multiple imputation by MICE is faster than by MIST. Other datasets and scenarios revealed the same behavior. MICE was also shown to be faster than MIST in [11]. The reason is that MIST uses non-parametric tree models with Bayesian bootstrap which can make it more difficult for the algorithm to converge, compared to the standard MICE that uses linear parametric models. Therefore, MICE + CondRF makes a better trade-off between performance and computational cost than MIST + CondRF. Only for complex datasets with strong nonlinear dependencies MIST + CondRF gives a better performance at the cost of a higher computation time.

Next to CondBagging, single imputation methods and procedures with surrogates have stable computational time across scenarios as well, but they may only work for small amounts of missing data. Only results of Prox. matrix + RF are shown in the plots. In general, methods show the lowest computation times under MAR and the largest times under MNAR mechanism. The latter is especially the case for methods with MI. This is not surprising since, given the complexity of the MNAR mechanism, methods with MI will need more time until they achieve convergence.

We also inspected the time evolution of the different techniques as sample size and/or dimension increases. In particular, we recorded the computation time for the experiments described in subsection 5.2. Figure 7 shows the time evolution in CPU minutes for datasets with 40% of the values MNAR on all features. When the sample size increases we note from Figure 7A that for almost all methods time increases quasi exponentially. Overall, CondBagging consistently has a

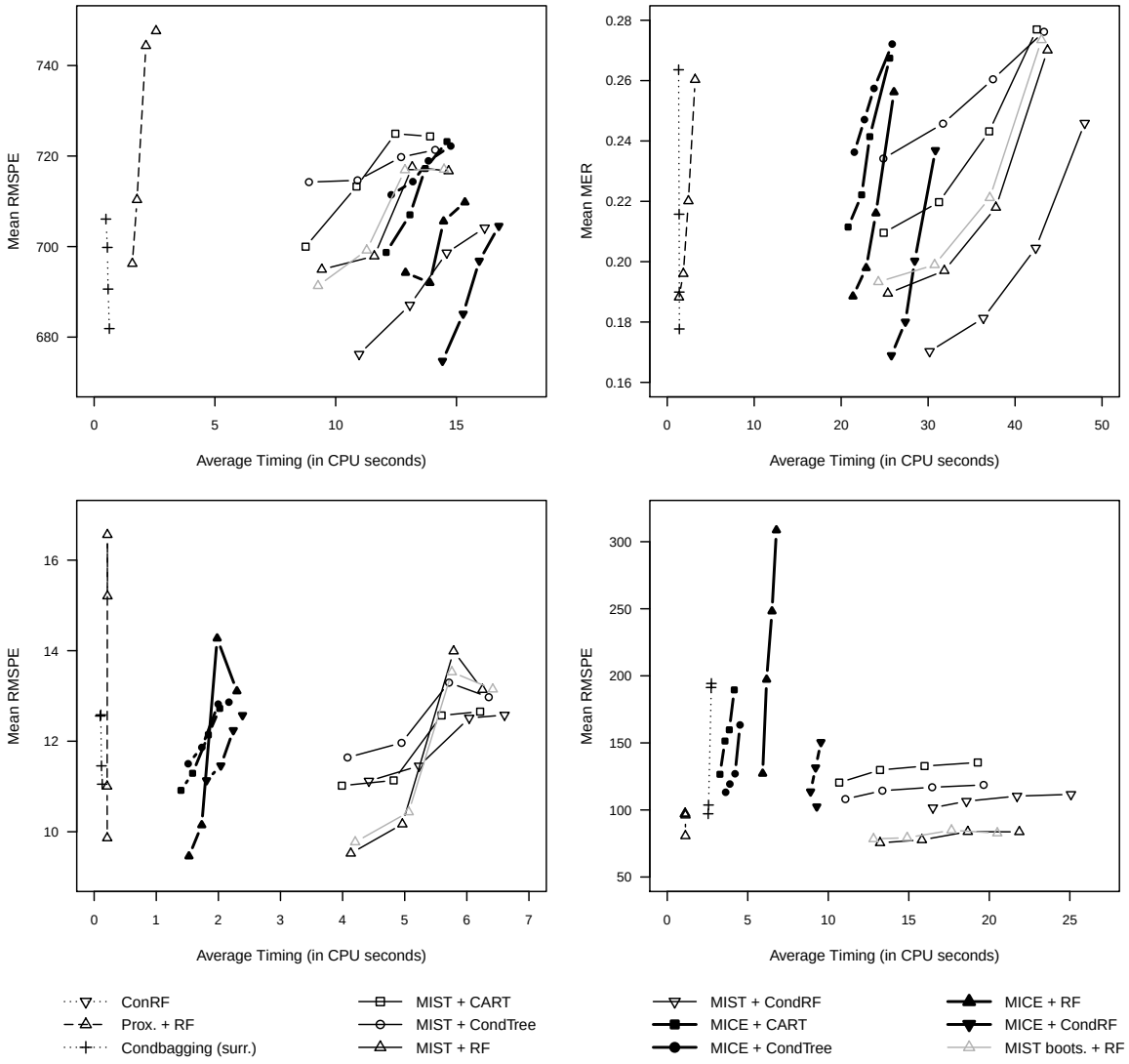

Figure 6: Performance vs computation time for the Birthweight (top left), Heart (top right), Fertility (bottom left) and simulated (bottom right) datasets. The mean RMSPE values for the Birthweight dataset have been divided by 10^5.

lower computation time compared to MICE/MIST + CondRF techniques, with the highest differences for MIST + CondRF. MICE is faster than MIST in all scenarios. In general techniques take longer to be computed under data MNAR (plots for MCAR and MAR are not shown here).

When the dimension grows (Figure 7B), the computation time of MICE/MIST methods clearly increases faster than for growing sample size, while CondBagging keeps a similar speed in both

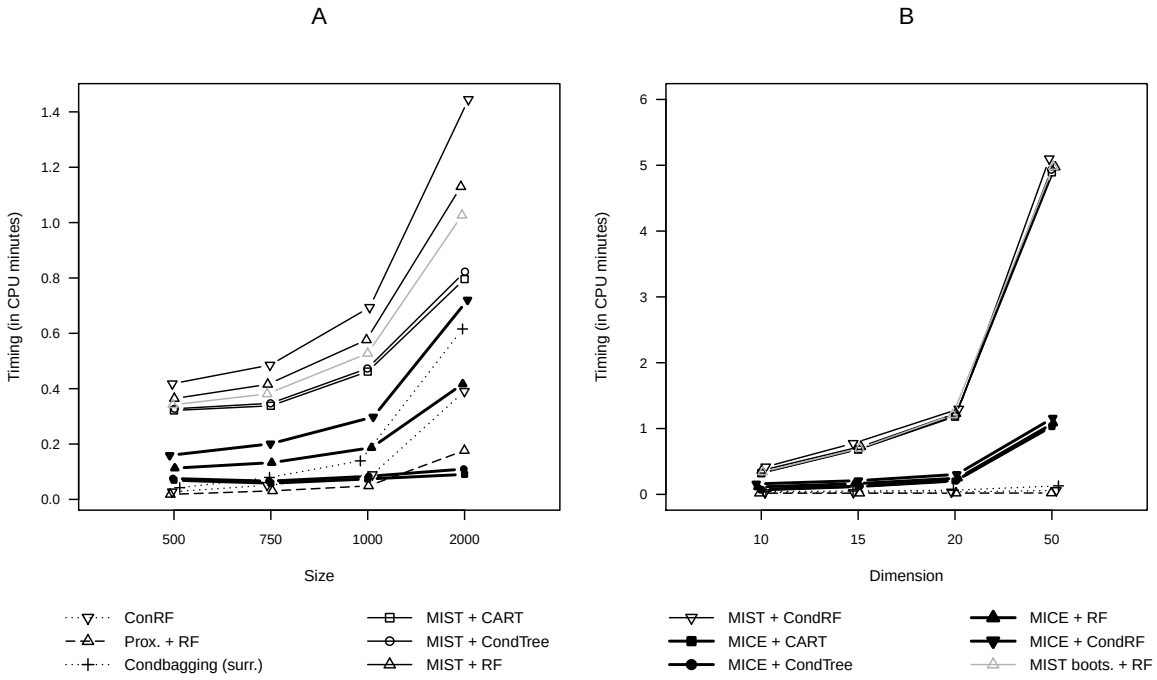

Figure 7: A: Computation time vs sample size for simulated data; B: Computation time vs dimension for simulated data

experiments. MICE is still faster than MIST and methods take longer to be computed under the MNAR mechanism. On datasets of size 1000 in 50 dimensions MIST combined with ensembles required the longest computation time, reaching even around 8 minutes. The increase in computation time as size and/or dimension grows could be expected, especially for the methods with MI. However, computing times of around 8 minutes on a standard machine are still manageable. Hence, MI + CondRF can be computed at a reasonable time even for larger datasets.

6 Conclusions and Future work

If in real-life applications the practitioner does not know the mechanism that generated the missing data, as often is the case in practice, we recommend the following strategies when building a prediction model using tree-based methods:

- If small amounts of missingness are present it suffices to apply any ensemble method with surrogates or with a previous single imputation.
- If the data contains moderate to large amounts of missing values, then multiple imputation by MICE or MIST followed by CondRF is the safest option.

- For high dimensional datasets, CondBagging with surrogate decisions yields a good compromise between performance and computation time.

Multiple imputation is preferred over single imputation because the latter often does not yield any improvement over surrogates. The new method of MIST imputed bootstrap samples when combined with RF is also not able to outperform MI. Multiple imputation ensembles in general showed good results in our comparisons, especially when the amount of missing data was large. These scenarios potentially lead to high prediction variability. Thus, it is crucial that the prediction rule has the ability to average out these sources of variability. Thanks to their ensemble nature in both the imputation step and the prediction model, MI ensembles are able to cancel out both the sampling variability and the variability caused by the missing data. Our theoretical derivations support the empirical findings. However, our studies showed that prediction performance of MI + RF may deteriorate compared to MI + CondRF for large fractions of data MNAR. Most likely the strategy used to select a splitting variable in each region by the individual conditional trees in CondRF allows to better extract valuable information from the imputations than RF in this complex scenario. Therefore, MI + CondRF is a safer and more robust method in terms of prediction performance.

In high dimensions, variability reduction by averaging in MI methods like MICE/MIST + CondRF can be exceeded by the extra noise introduced in the imputation process due to a large number of noisy predictors. In these cases CondBagging emerges as a very good and computationally cheaper alternative.

As with all empirical studies it is not possible to make broad generalizations of our results to other real-life settings. Our conclusions will be applicable to datasets of similar structure (correlations, size, dimension,...) when using the same settings for the tuning parameters of the methods as in our case (see Table 3). However, in our opinion these conclusions form a good reference more generally, as the various real-life datasets were selected from different scientific fields with variation in the number of observations and variables. Moreover, the artificial dataset was simulated with a very complex structure for its predictor variables and their relation with the outcome variable. The large comparison of several techniques across many missing data scenarios, which were at times extreme, also enriches the utility and relevance of this study as an element of reference.

In this study, we have combined missing data procedures with tree-based prediction methods in such a way that the whole procedure can first be learned on the training data and then be used to make predictions for individual test cases, when both contain missing values. In our evaluation of the techniques, we only considered complete test cases to avoid an extra source of variability in the performance measures. In principle, all methods considered in this study can handle test cases with missing values, but the currently available implementations in R are not flexible enough yet to obtain the predictions in practice. For example, for the imputation approaches it would be necessary that an incomplete test case can be imputed on the basis of the imputation model from the training data before entering the tree model. Implementations of imputation methods like mice() or rfImpute() currently do not have the feature to "predict" the missing data in a new case based on the imputation fit(s) of the training data. Therefore, current implementations need to be updated and extended with an associated predict function to make them applicable in practice.

Acknowledgements

Financial support from the Interuniversity Attraction Poles Programme (IAP-network P7/06), Belgian Science Policy Office, is gratefully acknowledged.

References

[1] A. Asuncion, D.J. Newman, UCI Machine Learning Repository, 2007.

[2] G.E. Batista, M.C. Monard, An analysis of four missing data treatment methods for supervised learning, in: Applied Artificial Intelligence, volume 17, pp. 519–533.

[3] L. Breiman, Bagging predictors, Machine Learning 24 (1996) 123–140.

[4] L. Breiman, Bias, Variance, and Arcing Classifiers, Machine Learning (1996).

[5] L. Breiman, Random forests, Machine learning 45 (2001) 5–32.

[6] L. Breiman, J. Friedman, R. Olshen, C. Stone, Classification and Regression Trees, Chapman and Hall, CRC, 1984.

[7] P. Bühlmann, B. Yu, Analyzing bagging, The Annals of Statistics 30 (2002) 927–961.

[8] L.F. Burgette, J.P. Reiter, Multiple imputation for missing data via sequential regression trees, American Journal of Epidemiology 172 (2010) 1070–1076.

[9] A.P. Dempster, N.M. Laird, D.B. Rubin, Maximum likelihood from incomplete data via the EM algorithm, Journal of the Royal Statistical Society B 39 (1977) 1–38.

[10] T.G. Dietterich, E.B. Kong, Machine Learning Bias , Statistical Bias , and Statistical Variance of Decision Tree Algorithms, Machine Learning 255 (1995) 0–13.

[11] L.L. Doove, S. Van Buuren, E. Dusseldorp, Recursive partitioning for missing data imputation in the presence of interaction effects, Computational Statistics and Data Analysis 72 (2014) 92–104.

[12] B. Efron, Bootstrap Methods: Another Look at the Jackknife, The Annals of Statistics 7 (1979) 1–26.

[13] A.J. Feelders, Handling missing data in trees: surrogate splits or statistical imputation, in: PKDD'99: Proceedings of the Third European Conference on Principles of Data Mining and Knowledge Discovery, volume 1704 of *Lecture Notes in Computer Science*, Springer-Verlag, London, UK, 1999, pp. 329–334.

[14] S. Geman, E. Bienenstock, R. Doursat, Neural Networks and the Bias/Variance Dilemma, 1992.

[15] A. Hapfelmeier, T. Hothorn, K. Ulm, Recursive partitioning on incomplete data using surrogate decisions and multiple imputation, Computational Statistics & Data Analysis 56 (2012) 1552–1565.

[16] A. Hapfelmeier, K. Ulm, Variable selection by Random Forests using data with missing values, Computational Statistics & Data Analysis 80 (2014) 129–139.

[17] T. Hastie, R. Tibshirani, J. Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Second Edition., Springer New York, 2009.

[18] Y. He, Missing Data Imputation for Tree-Based Models, Ph.D. thesis, University of California, Los Angeles, 2006.

[19] N.J. Horton, K.P. Kleinman, Much ado about nothing: a comparison of missing data methods and software to fit incomplete data regression models., The American Statistician 61 (2007) 79–90.

[20] T. Hothorn, K. Hornik, C. Strobl, A. Zeileis, Party: a laboratory for recursive part(y)itioning, R Package Version 0.9-99996 (2011).

[21] T. Hothorn, K. Hornik, A. Zeileis, Unbiased Recursive Partitioning: A Conditional Inference Framework, Journal of Computational and Graphical Statistics 15 (2006) 651–674.

[22] A. Kapelner, J. Bleich, Prediction with Missing Data via Bayesian Additive Regression Trees, arXiv.org stat.ML (2013).

[23] M.A. Klebanoff, S.R. Cole, Use of multiple imputation in the epidemiologic literature., American Journal of Epidemiology 168 (2008) 355–357.

[24] S.G. Liao, Y. Lin, D.D. Kang, D. Chandra, J. Bon, N. Kaminski, F.C. Sciurba, G.C. Tseng, Missing value imputation in high-dimensional phenomic data: imputable or not, and how?, BMC bioinformatics 15 (2014) 346.

[25] A. Liaw, M. Wiener, Classification and regression by randomForest, R News 2 (2002) 18–22.

[26] R.J.A. Little, D.B. Rubin, Statistical Analysis with Missing Data, 2002.

[27] G. Louppe, Understanding random forests from theory to practice, Ph.D. thesis, University of Liege, 2014.

[28] R.J. Marshall, P. Kitsantas, Stability and Structure of CART and SPAN Search Generated Data Partitions for the Analysis of Low Birth Weight, Journal of Data Science 10 (2012) 61–73.

[29] A. Peters, T. Hothorn, B. Lausen, ipred: Improved predictors, R News 2 (2002) 33–36.

[30] J.R. Quinlan, C4.5: Programs for Machine Learning, volume 1, 1993.

[31] R Development Core Team, R: a language and environment for statistical computing, Technical Report, Vienna, Austria, 2011.

[32] A. Rieger, T. Hothorn, C. Strobl, Random Forests with Missing Values in the Covariates, Technical Report 79, Ludwig-Maximilians-Universität Munich, Germany, 2010.

[33] D.B. Rubin, The Bayesian Bootstrap, The Annals of Statistics 9 (1981) 130–134.

[34] D.B. Rubin, Multiple Imputation for Nonresponse in Surveys, John Wiley & Sons, New York, 1987.

[35] D.B. Rubin, Multiple imputation after 18+ years, Journal of the American Statistical Association 91 (1996) 473–489.

[36] M. Saar-Tsechansky, F. Provost, Handling Missing Values when Applying Classification Models, Journal of Machine Learning Research 8 (2007) 1625–1657.

[37] J.L. Schafer, Analysis of Incomplete Multivariate Data, Chapman & Hall, 1997.

[38] A.D. Shah, J.W. Bartlett, J. Carpenter, O. Nicholas, H. Hemingway, Comparison of random forest and parametric imputation models for imputing missing data using MICE: A CALIBER study, American Journal of Epidemiology 179 (2014) 764–774.

[39] D.J. Stekhoven, P. Bühlmann, Missforest-Non-parametric missing value imputation for mixed-type data, Bioinformatics 28 (2012) 112–118.

[40] M.A. Tanner, W.H. Wong, The Calculation of Posterior Distributions by Data Augmentation, Journal of the American Statistical Association 82 (1987) 528–540.

[41] T.M. Therneau, B. Atkinson, rpart: recursive partitioning, R Package Version 3.1-50; R Port by B. Ripley (2011).

[42] R. Tibshirani, Bias, variance and prediction error for classification rules, Monographs of the Society for Research in Child Development 79 (1996) 1–14.

[43] O. Troyanskaya, M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, D. Botstein, R.B. Altman, Missing value estimation methods for DNA microarrays., Bioinformatics 17 (2001) 520–525.

[44] W. Vach, Logistic Regression with Missing Values in the Covariates, Springer New York, 1994.

[45] S. Van Buuren, Flexible Imputation of Missing Data, Chapman and Hall/CRC 2012, 2012.

[46] S. Van Buuren, J.P. Brand, C.G. Groothuis-Oudshoorn, D.B. Rubin, Fully conditional specification in multivariate imputation, Journal of Statistical Computation and Simulation 76 (2006) 1049–1064.

[47] S. Van Buuren, K. Groothuis-Oudshoorn, MICE: Multivariate Imputation by Chained Equations in R, Journal of Statistical Software 45 (2011) 1–67.

[48] I.R. White, J.B. Carlin, Bias and efficiency of multiple imputation compared with complete-case analysis for missing covariate values, Statistics in Medicine 29 (2010) 2920–2931.